Behavioral Anomaly Detection in Forensics Analysis

Journal:	*IEEE Security & Privacy*
Manuscript ID	SPSI-2017-03-0081.R1
Manuscript Type:	SPSI: Digital Forensics
Date Submitted by the Author:	26-May-2017
Complete List of Authors:	Al-Haj Baddar, Sherenaz; University of Jordan, Merlo, Alessio; University of Genoa, DIBRIS Migliardi, Mauro; University of Padua, Italy, Dipartimento di Ingegneria dell'Informazione
Keywords:	anomaly detection, outlier detection, forensic analysis, profiling, sketching

Behavioral Anomaly Detection in Forensics Analysis

Sherenaz Al Haj Baddar, Alessio Merlo *Member, IEEE,* and Mauro Migliardi

***Abstract*—In cybercrimes pertaining to networking activities, forensics activities and user privacy rights are often competing forces as it is illegal to examine messages contents without a court warrant. Furthermore, examination without users permission may be impossible if cryptography has been used. Thus, lightweight forensics tools, capable of providing a first line of warning without infringing users privacy are needed. We argue that packet summarization, combined with dynamic anomaly detection, could provide evidences sufficient to identify malicious actor(s) within a network. Moreover, when an evidence that entities are straying from normal behavior exists, without breaching their privacy, legal access to their messages contents becomes possible. Our contribution in the field of lightweight forensics tools is twofold: first we introduce CATTURE, a lightweight statistical detector capable of identifying behavioral anomalies among network nodes timely, without jeopardizing users' privacy; second, we leverage the expertise of network administrators to ground statistical behavioral anomalies, and cite related events to reduce false positives and increase the sensitivity of the system towards false negatives.**

***Index Terms*—anomaly detection, outlier detection, forensic analysis, profiling, sketching**

I. Introduction

Cybercrime is on the rise, yet network forensics still struggle with the problem of identifying misdemeanors without compromising innocent users' privacy. In fact, while looking into a user data stream may in some cases provide incriminating information, in most countries it is illegal to do so without an explicit warrant. Moreover, recent studies on digital forensics cite several challenges among which is the lack of effective instant anomaly detection tools, which deprives forensic analysis tools from initial leads vital for their operations (see, e.g., [9], [7], [14]). Hence, it is necessary to adopt a layered approach where forensics tools both lightweight and privacy preserving are deployed to provide a first level of detection for suspicious behaviors. When such a behavior is detected, further investigation will be deployed targeting a limited subset of the traffic with the twofold advantage of reducing the computational requirements and, if necessary, providing the basis for the request of an actual warrant. We introduce CATTURE, a lightweight, privacy-preserving, behavioral anomaly detection tool that does not assume any prior knowledge on what constitutes an anomaly. The CATTURE investigation is based only on anonymized traffic features and its results respect the legal constraints defined by the EU for the publication of network traffic. Furthermore, we show how it is possible to leverage the specific knowledge and expertise of the network administrators of a specific site in order to ground statistical signatures and statistically anomalous behaviors to specific unseasonal events in order to cull the number of false positives without reducing the sensitivity of the system and generating false negatives.

After generating packets statistical summaries in terms of a pre-defined set of features for all nodes observed, CATTURE generates clusters according to each feature then aggregates nodes with similar patterns together forming profiles. In order to identify malicious activities, outlier detection is deployed to single out profiles with extreme feature values, relatively speaking, and label the nodes comprising them as anomalous.

One of the most problematic aspects of anomaly detection solutions is their inability to cope with special events and ephemeral modifications without generating a flurry or false positives [4], [5]; at the same time, the sensitivity to changes in behavior that are specifically dangerous or suspicious are often treated not differently from any other change and thus can lead to very troublesome false negatives. For these reasons, we have decided to "ground" our statistical analysis to the actual meaning of events by fusing the wisdom of network experts, actually, the expertise of the system managers of the sites where our tool has been tested. This way, we enrich our statistical anomaly detection system with expert-knowledge reducing the frequency of both false positives and false negatives.

Structure of the paper. Section 2 highlights related work, while Section 3 presents the CATTURE architecture and describe its operations in details. Section 4 reports the experiments conducted using an actual dataset collected from a campus network. It also illustrates the results obtained considering the insights provided by network administrators. Finally, the conclusions and future work of this study are depicted in Section 5.

II. Related Work

Recent research on forensics analysis has opted for applying intrusion detection solutions to help identify malicious parties pro-actively. Here we shed light on some state-of-the-art examples, like the Network Forensics based on Intrusion Detection Analysis (NFIDA) system [6]. This system utilizes packet headers to perform offline analysis that comprises pattern matching and protocol analysis and utilizes library traces in order to identify packets that convey malicious behavior. A hybrid attack detection and anti-honeypot-based forensics model

Sherenaz Al Haj Baddar is at The University of Jordan, Amman, Jordan, Email: s.baddar@ju.edu.jo.

Alessio Merlo is at DIBRIS, University of Genoa, Italy, Email: alessio@dibris.unige.it.

Mauro Migliardi is at DEI, University of Padua, Italy. Email: mauro.migliardi@unipd.it.

addressing DDoS attacks in Machine-to-Machine networks, HADFM, is depicted in [13]. The proposed solution utilizes a decentralized intrusion detection framework that reacts to detected threats in a real-time fashion. The Alert Detection Expert System (ADES) forensics tool comprises intrusion detection as depicted in [8]. ADES analyzes large amounts of logs to identify DDoS attacks pro-actively using a combination of Shannon-entropy concept and clustering algorithms. The offline centralized forensics system depicted in [1], Admissible Network Forensics Correlation Model (ANFCM), utilizes a log-based intrusion detection component, then applies decision tree algorithms in order to filter anomalous behaviors. Afterwards, ANFCM re-routes the logs to a central repository where event-logs management functions are applied. The Fuzzy Logic-based System for Origin of Attack Detection (FLSOAD) is a network forensics tool that aims at identifying the origin of attack using intrusion detection as depicted in [10]. It comprises an expert system based on fuzzy logic to identify the time, origin, and method of the attack. Behavioral anomaly detection is also used in the network forensics with Dempster-Shafer theory tool, NFDST, proposed in [12]. NFDST is a decentralized digital evidence fusion tool that utilizes efficient data mining and reasoning theory techniques to detect and fuse anomalous behavior from different sources in a real-time manner.

As depicted in these examples, network forensics analysis tools that utilize intrusion detection vary with respect to their processing style, thus while some operate in a centralized fashion, others are distributed. Also, forensics tools responsiveness differs, as some provide real-time responses as soon as abnormalities surface, as others digest the collected information and provide feedback offline. As for the approach forensics tools deploy, it spans machine learning and information theoretic techniques among other approaches. The end-goal of a forensics tool differs as well; some tools aim at identifying the origin of an attack, while others aim at visualizing the attack. Also, some forensic tools aim at reconstructing the attacks while others aim at analyzing intrusion data. Table I depicts a taxonomy of network forensic tools.

III. The CATTURE Architecture

The Cluster-based Anomaly deTection using skeTURE engine (CATTURE) is an anomaly detection system targeted at identifying network nodes with anomalous behavior without jeopardizing users' privacy. It also aims at providing an understanding of the underlying network behavior by mathematically modeling the operations of the nodes comprising the designated network.

In order to achieve its goal, CATTURE analyzes certain features of the network traffic, using the sketch-based tool, SKETURE, then applies an efficient clustering technique to categorize the nodes within the network. Afterwards, CATTURE applies a profiling approach to generate sets of patterns that describe the behavior of the nodes. Then, it compares these sets of patterns in order to identify the nodes that are deviating from normal behavior. The CATTURE system comprises three modules: i) the *packet-analysis module*, ii) the *profiling module*, and iii) the *anomaly detection module*. We now describe the operations of these modules, which are depicted in Fig. 1.

A. The Packet Analysis Module

In order to model the behavior of a given node, its incoming and outgoing packets need to be analyzed. CATTURE utilizes SKETURE [3], the sketch-based packet analysis tool, for packet analysis. As it splits time into equal intervals, SKETURE examines the headers of the packets after obfuscating their address information, and builds a statistical summarization of each node n's behavior in terms of a pre-selected set of features during each such interval [3]. In this study, we chose the following features in order to summarize the behavior of nodes:

- The packet size, in bytes
- The packet count
- The number of unique destinations
- The total amount of bytes

B. The profiling Module

The profiling module in CATTURE comprises two phases; the clustering phase, and the reduction phase. The clustering phase clusters the nodes according to each feature separately, while the reduction phase aggregates the nodes with similar clustering across different features into profiles. In this subsection we describe these two phases in detail.

1) The Clustering Phase: Clustering in CATTURE happens at two levels; at the first level we cluster nodes according to how long they have been active in the network, i.e., we apply duration-based clustering, and label these clusters as top-groups. At the second level, we cluster the nodes within each top-group according to the values of their corresponding features, i.e., we apply feature-based clustering. The duration-based clustering in CATTURE creates three top-groups; the *Short-Lived* top-group denoted by $\mathcal{S}$, the *medium-lived* top-group denoted by $\mathcal{M}$, and the *long-lived* top-group denoted by $\mathcal{L}$. While top-group $\mathcal{S}$ contains the nodes that remained active for the fewest time intervals, the $\mathcal{L}$ group contains the nodes that were active for the most time intervals, whereas remaining nodes get assigned to the $\mathcal{M}$ top-group. We chose to do this duration-based clustering before analyzing nodes' behaviors because it is unlikely that nodes that appear sporadically will behave like nodes that are active almost all the time. Next, feature-based clustering executes for each top-group separately. Thus, within each top-group, CATTURE applies a divisive clustering algorithm, QUIST [2], considering each feature separately. Considering a given top-group d as an initial cluster with respect to a given feature f, CATTURE calculates the overall average of node n values. Then, it sorts the set of all such values across all nodes in d. After that, it applies the QUIST *spreadness* metric which estimates the scatterness of values within a set and determines whether the nodes within the set need to be split into further sub-clusters or not. If the *spreadness* value of the initial cluster exceeds a pre-set threshold, the cluster is split into two parts at its median. This process is repeated iteratively until each remaining cluster has

TABLE I
TAXONOMY OF FORENSICS ANALYSIS TOOLS.

Forensics Tool	Processing	Responsiveness	Approach	Objective
CATTURE	centralized	real-time	machine learning	origin of attack
NFIDA [6]	centralized	offline	machine learning	analysis of intrusion data
HADFM [13]	decentralized	real-time	hybrid	reliable forensic evidence
ADES [8]	centralized	real-time	hybrid	reliable forensic evidence
ANFCM [1]	centralized	offline	decision trees	reliable forensic evidence and attack reconstruction
FLSOAD [10]	centralized	real-time	fuzzy logic	origin of attack
NFDST[12]	decentralized	real-time	hybrid	reliable forensic evidence

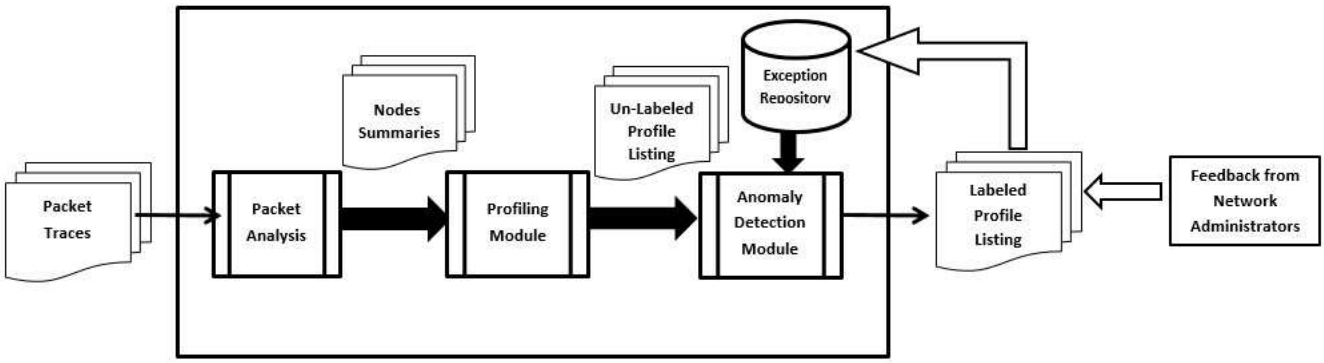

Fig. 1. The CATTURE Architecture

fewer nodes than a pre-defined size s, or has a *spreadness* value below the pre-set threshold. QUIST also terminates when the number of generated clusters reaches a pre-defined limit K, whichever happens first. After creating separate clusters of all features within each top-group d, the next phase reduces the generated clusters into profiles.

2) The Reduction Phase: In this phase, CATTURE designates each cluster generated with respect to feature f, C^f, by an interval comprising the minimum and maximum values that it contains, denoted by $[min_{C^f}, max_{C^f}]$. Then, it examines the clusters to which each node belongs with respect to each feature f, and creates a corresponding association, which depicts each cluster to which a given node belongs with respect to each feature f. Next, all nodes with identical associations are assigned to a profile, p_i. However, this may generate numerous profiles; thus, CATTURE merges similar profiles. More precisely, as profiles comprise associations, profiles p_i and p_j are similar if their corresponding associations are similar for each feature f. Also, two associations are considered similar with respect to feature f if all the respective intervals comprising them are similar. Two intervals are similar if either one is contained in the other, or if most of their values are in common. Furthermore, if the intervals are too close to each other, i.e. the distance between the two furthest endpoints of two intervals is less than or equal to a tolerance constant, then they are considered similar.

Thus, if the associations of two profiles are similar, then they are considered similar and are thus merged into a new profile that contains all the nodes that appeared in both profiles originally. The process of reducing similar profiles continues until no profiles are similar.

C. The Anomaly Detection module

This module applies statistical outlier detection to isolate outlier profiles with respect to each feature f. The technique adopted is modified Z-Score, which comprises a standardized score that measures how much a given value differs from typical values within a set of values[1] . We run this statistical test across all profiles within a given top-group, d, with respect to a particular feature f. If a profile passes the test, we label it as normal with respect to feature f, otherwise, we label it as abnormal with respect to feature f. If a given profile is labeled normal with respect to all features f, then it is labeled as *normal*. Otherwise, it is labeled as potentially *Outlier*. Consequently, nodes that comprise that profile are also labeled as potentially malicious.

This labeling serves as a first-level alarm, yet, it may convey false alarms. Therefore, CATTURE checks its *Exception repository* for exceptions provided by the network administrator; if a profile it labeled appears with a different labeling in the repository, it fixes the label accordingly.

Administrators may also examine nodes within profiles CATTURE labeled as *Outlier* and provide further feedback. Upon doing so, the network administrator would either confirm the CATTURE labeling of a given potentially *Outlier* profile, or alternatively add that profile to CATTURE's *Exception repository*.

[1]https://www.ibm.com/support/knowledgecenter/en/SSWLVY_1.0.1/com.ibm.spss.analyticcatalyst.help/analytic_catalyst/modified_z.html

IV. Experimentation and Results

In this section we present the dataset we used for assessing CATTURE alongside the CATTURE implementation. Then, we depict the preliminary results and highlight the insights obtained from domain experts.

A. The Padua Dataset

The Padua dataset comprises the traffic of the Department of Information Engineering (DEI) at the University of Padua over a month starting on April 5th 2015, and ending on May 5th, 2015 [3]. In order to perform our studies without jeopardizing users' privacy and to comply with the EU regulations, our traces only comprised the following information per packet:

- obfuscated IP addresses
- packet timestamp
- packet length

B. Implementation

The implementation of CATTURE is based both on the SKETURE tool depicted in [3] and on an implementation of the profiling and anomaly detection modules in Java v. 8.0. SKETURE was first used to generate per-minute summaries for the traces collected on Monday April 27^{th} 2015, Friday May 1^{st} 2015, and Sunday May 3^{rd} 2015 each aside. Then, the profiling module performed a first-level clustering that generated 3 top-groups for Short-Lived, Medium-Lived, and Long-Lived nodes. The Short-Lived group spanned nodes that were alive for less than ten minutes, and comprised most of the nodes, while Medium-Lived nodes were active for up to 20 minutes and comprised only 6% of nodes observed. Moreover, Long-Lived nodes were active for more than 20 minutes and up to 24 hours, and comprised around 15% of nodes observed. We ran our experiments without utilizing an exception repository as we have not had any pre-insights on what would constitute an exception at the time.

C. Preliminary Results

In this subsection we summarize the labeled profiles CATTURE generated for the Monday April 27^{th} traces which, according to the DEI network administrators, was rather a regular working Monday on campus. We also contrast the behavior exhibited by the network on that day with its behavior on Friday May 1^{st} which is an official Holiday in Italy and with Sunday May 3^{rd} which is a typical weekend day. Finally, we highlight time and memory requirements of the profiling and anomaly detection modules in CATTURE.

1) Network Behavior: Comparing the number of nodes observed in 3 different days, namely Monday April 27^{th}, Friday May 1^{st} and Sunday May 3^{rd}, we notice that Monday had the highest number of nodes. In fact, an official holiday (May 1st) and a Sunday had only 54% of the nodes observed on the Monday.

On the other hand, analyzing the number of profiles generated for each of the examined days, we observe that while Monday April 27^{th} had 98 profiles, Friday May 1^{st}, although an official holiday, had a close number of profiles, 92, while Sunday as a weekend only had 83 profiles. Furthermore, nodes that remained active on Friday were almost 50% of the nodes reported on Monday; however, they produced 92 different profiles. On the other hand, Sunday had a similar number of nodes like Friday, yet, it had less profiles. This indicates that network administrators probably had the nodes do some routine tasks on Friday which they would not normally do on a Sunday.

To gain a better understanding of how the observed nodes behaved, we now illustrate a summary of the labeled profiles exhibiting normal and outliers based on statical detection as identified by CATTURE. While none of the reported packet sizes in Short-Lived profiles on Monday were statistically abnormal, 8% of Medium-Lived profiles had abnormal packet sizes that ranged from almost 0.8KB up to almost 3KB. Moreover, 28.2% of Long-Lived nodes exhibited statistically abnormal packet sizes ranging between about 0.3KB up to around 3.4KB. As network administrators asserted there were no abnormalities reported that day, we will consider these statistical outliers as false positives. Yet, should we have had a priori knowledge on nodes roles, we could have identified anomalies that were missed by the network administrators.

We now analyze the statistical analysis of packets count in relation to duration clustering. First, only 0.26% of Short-Lived nodes exhibited statistically anomalous behavior in terms of packet counts, as they sent from almost 8.5K packets to 140K packets per minute; at the same time, 7.5% of Medium-Lived had statistically abnormal number of packets per minute as almost 6.7% of them sent up to 6.5K packets per minute, while almost 0.8% of them sent up to 53.5K packets. As for Long-Lived nodes only 0.7% of them exhibited statistically abnormal packet counts reaching up to 231K packets per minute. Again, as network administrators stated that they have not noticed any abnormalities on Monday April 27^{th}, we will consider the CATTURE outliers false positives. However, the existence of priori knowledge on nodes roles should have helped us identify stealthy anomalies not spotted by network administrators.

With reference to the number of unique destinations of packets we observe that none of the Short-Lived nodes contacted more than 2304 different destinations per minute, while 36.15% of Medium-Lived nodes contacted at most one unique destination per minute. On the other hand, 0.72% of Long-Lived nodes contacted at most 390 unique destinations. As CATTURE classified the Medium-Lived nodes with at most one unique destination per minute together with Long-Lived nodes with more than 20 unique destinations per minute as statistically anomalous, we still cannot confirm that either one of those nodes is actually anomalous unless we have further information on their position in the network and the types of tasks they were expected to do.

The statistical analysis of the total bytes transmitted by nodes clustered according to their duration shows that about 6% of the Short-Lived nodes exhibited a statically anomalous behavior by sending more than almost 0.2MB and up to 33MB of data per minute. Also, almost 15% of Medium-Lived nodes sent more than 69KB and up to about 80MB of data per minute, and were flagged as statistical anomalies.

As for Long-Lived nodes, almost 18% of them exhibited statistically anomalous behavior by sending more than 33KB and up to about 342MB of data per minute. Aside from being statistically abnormal, network administrators would probably be interested in investigating such nodes to validate their operations when such an observation is generated by CATTURE.

We also compared the behavior exhibited by the nodes on Monday, Friday, and Sunday, with respect to the maximum value reported in each association across the 3 days. The maximum average packet size reported on Friday and Sunday exceeded the corresponding value reported on Monday for Short-Lived nodes. While the maximum average packet size reported in Short-Lived nodes on Monday was a bit below 4 KB, the maximum average packet size for Short-Lived nodes on Friday and Sunday reached almost 5KB. According to the network managers, this is probably due to the fact that more machine-driven automated traffic was generated on Friday and Sunday, compared to Monday, when human users were around. Furthermore, while Long-Lived nodes behaved almost similarly in the 3 days in terms of their packet sizes, Medium-Lived nodes sent much smaller packets on Sunday compared to Monday and Friday; on Sunday, Medium-Lived nodes sent packets each of no more than 1.5KB, compared to Monday and Friday were Medium-Lived nodes sent packets with sizes almost twice as large.

Analyzing the behavior in the three days with respect to packet counts, we observed that Short-Lived nodes on Monday sent more packets; in fact, their maximum packet count reached almost 150K packets per minute, while much less activity was observed on Friday and Sunday where packets sent per minutes did not exceed 80K. As Monday was a working day compared to an official holiday and a weekend, this behavior is rather expected. Yet, when Medium-Lived nodes are examined a different pattern is identified. Compared to Medium-Lived nodes on Monday that sent almost 50K packets per minute, and Medium-Lived nodes on Sunday which sent almost no more than 20K packets per minute, Medium-Lived nodes on Friday peaked them both by sending almost 400K packets per minute. This is quite interesting as Friday was an official holiday, thus, unless network administrators were running pre-scheduled tasks that would cause Medium-Lived nodes to send this much packets, this behavior implies an anomaly in the 30% of these Medium-Lived nodes and further investigation has been suggested.

We also compare the maximum number of unique destinations nodes contacted per minute. In the 3 days we are considering, Short-Lived nodes contacted almost 2200 different destinations per minute, while Medium-Lived nodes contacted no more than 500 unique destinations in the 3 days. As for the Long-Lived nodes, they contacted less than 500 unique destinations per minute on Monday, and almost 100 unique destinations per minute on Sunday. However, albeit being a day-off, some Long-Lived nodes contacted almost 1200 unique destinations on Friday. Again, unless network administrators can assert that this behavior was normal due to some pre-scheduled operations, one may suspect that these nodes were involved in a malicious behavior and further investigation is suggested.

Medium-Lived nodes also exhibited a different-than-expected behavior on Friday compared to Monday and Sunday. Some Medium-Lived nodes sent almost 600MB of traffic per minute on Friday, while on Monday they did not exceed 100MB of traffic per minute. Also, Medium-Lived nodes on Sunday did not send more than 4MB per minute. This further confirms our previous notes on Medium-Lived nodes behavior. As for Short-Lived nodes, their behavior on Monday compared to Friday and Sunday is rather expected; Short-Lived nodes on Monday sent up to 200 MB of data per minute, while their counterparts on Sunday and Friday did not exceed 100MB per minute. As for Long-Lived nodes, they did not send more than 9.6MB per minute on Sunday, and almost no more than 200MB on Friday, but as expected, they sent around 350MB of traffic per minute on Monday.

The outliers identified by CATTURE were considered false positives as network administrators stated that they have not identified any malicious behavior on Monday April 27^{th}. Nevertheless, the number of outlier nodes CATTURE recognized was way smaller than the actual network size, and the nodes worrisome features were clearly identified. Thus, the introduction of CATTURE allows adopting a second level of analysis with reduced computational and privacy infringement requirements.

2) Performance of CATTURE: If CATTURE fails at raising the first-level alarm timely, malicious behavior could go unnoticed until it is too late. In our experiments, while generating labeled profiles for 28MB of summaries required 5 seconds, it took almost 25 seconds to generate such outcomes for 168MB of summaries. As for memory consumption, the same set of experiments showed that, while running CATTURE for 28MB of summaries required 500MB of RAM, running it for 6 times more input increased the memory required to only 600MB. Hence we conclude that CATTURE is CPU hungry and the computational requirements are almost linear with the amount of data processed; however, the memory consumption pattern shows a significant saturation effect.

D. Experts Insights and Discussion

We discussed the results and outlier labels we obtained with the network administrators at the University of Padua. Aside from asserting that they are not aware of any abnormal behaviors on Monday April 27^{th}, they expressed their concern from having zombie nodes in their network. They also said that having an orchestrated set of nodes that do a designated action at a designated time is another source of concern. This may imply that the majority of the outliers identified by CATTURE as depicted in this section are false-positives, yet, this does not eliminate the chance that some nodes were actually abnormal. Discussion with network administrators confirmed our approach of first doing a durational-clustering and generate 3 top-groups. They said this would help them better contrast nodes behaviors. Moreover, network administrators re-emphasized the importance of tuning behavioral anomaly detection solutions to the environment in which they operate. Otherwise, statistical outliers may become irrelevant to nodes

behavioral context. This tuning also helps ground normal profiles and continuously improve the exception repository. Identifying a priori behavioral groups and generating a priori tags for each node may help grounding the statistical analysis. As an example, in the DEI network it is possible to identify three main behavioral groups: personal machines of staff members; public machines for students; and servers. Among the servers group further distinction may tag computational servers differently from web-services servers. Intuitively, it is obvious that these categories are too broad to allow pinpointing a single node in the set and thus breach anonimity; nonetheless, to guarantee the desired level of privacy, it is possible to design the tagging so that is guarantees k-anonimity [11]. Adopting this set of a priori tags would enrich the statistical analysis performed by CATTURE.

V. Conclusions and Future Work

In order to provide initial leads essential to effective investigations, forensic analysis tools need to overcome the problems of users' privacy and of immediate response times. For these reasons, in this paper we introduced CATTURE, a lightweight statistical detector capable of identifying behavioral anomalies both in time and among the network nodes population while complying with the privacy rules of Italian law, one of the most restrictive of the European Union in the field of privacy. CATTURE not only uses statistical outlier detection to identify anomalies, but also leverages domain experts' knowledge to help further refine its findings.

To evaluate the performance of CATTURE, we had it summarize and analyze the packet traces of a campus network on 3 days, Monday April 27^{th}, Friday May 1^{st}, and Sunday May 3^{rd} 2015. After building profiles of the nodes behaviors, we applied a statistical outlier detection technique to identify potentially malicious nodes. We contrasted the behavior of the network on the 3 days and described the outliers identified, yet, as the data were completely anonymized and the analysis was performed post-mortem, it was not possible to perform further investigations to identify the exact type of misdemeanor. We then discussed profiles and labels generated by CATTURE with network experts who suggested that labeling the nodes with an a priori behavioral category general enough to avoid privacy infringement (e.g., a web server vs. a desktop machine) could have helped identify genuinely malicious nodes. Finally, we depicted time and memory requirements of CATTURE on dataset samples of varying sizes.

We will improve CATTURE to address a set of issues highlighted by the domain experts like including a calendar-themed exception repository that would allow CATTURE to recognize when certain exceptions are likely to happen. Moreover, we will introduce node tags that pre-assign roles to nodes a priori, in order to pinpoint nodes deviating from normal behavior without jeopardizing users' privacy.

VI. Acknowledgements

The collection of data has been successful because of the cooperation of the personnel at the *Dipartimento di Ingegneria dell'Informazione of the University of Padua.* We specially thank Mr. Marco Filippi for his extremely positive attitude and his willingness to apply his script debugging capabilities to our project.

References

[1] A. Al-Mahrouqi, S. Abdalla, and T. Kechadi. Efficiency of network event logs as admissible digital evidence. In *2015 Science and Information Conference (SAI)*, pages 1257–1265, July 2015.

[2] S. W. A. Baddar. Short communication on QUIST: A quick clustering algorithm. *CoRR*, abs/1606.00398, 2016.

[3] S. W. A. Baddar, A. Merlo, and M. Migliardi. Generating statistical insights into network behavior using SKETURE. *J. High Speed Networks*, 22(1):65–76, 2016.

[4] M. H. Bhuyan, D. K. Bhattacharyya, and J. K. Kalita. Network anomaly detection: Methods, systems and tools. *IEEE Communications Surveys Tutorials*, 16(1):303–336, First 2014.

[5] V. Chandola, A. Banerjee, and V. Kumar. Anomaly detection: A survey. *ACM Comput. Surv.*, 41(3):15:1–15:58, July 2009.

[6] L. Jiang, G. Tian, and S. Zhu. Design and implementation of network forensic system based on intrusion detection analysis. In *2012 International Conference on Control Engineering and Communication Technology*, pages 689–692, Dec 2012.

[7] S. Khan, A. Gani, A. W. A. Wahab, M. Shiraz, and I. Ahmad. Network forensics: Review, taxonomy, and open challenges. *Journal of Network and Computer Applications*, 66:214 – 235, 2016.

[8] A. Olabelurin, S. Veluru, A. Healing, and M. Rajarajan. Entropy clustering approach for improving forecasting in ddos attacks. In *2015 IEEE 12th International Conference on Networking, Sensing and Control*, pages 315–320, April 2015.

[9] E. S. Pilli, R. Joshi, and R. Niyogi. Network forensic frameworks: Survey and research challenges. *Digital Investigation*, 7(1–2):14 – 27, 2010.

[10] M. Rostamipour and B. Sadeghiyan. Network attack origin forensics with fuzzy logic. In *2015 5th International Conference on Computer and Knowledge Engineering (ICCKE)*, pages 67–72, Oct 2015.

[11] P. Samarati and L. Sweeney. Protecting privacy when disclosing information: k-anonymity and its enforcement through generalization and suppression. Technical report, 1998.

[12] Z. Tian, W. Jiang, Y. Li, and L. Dong. A digital evidence fusion method in network forensics systems with dempster-shafer theory. *China Communications*, 11(5):91–97, May 2014.

[13] K. Wang, M. Du, Y. Sun, A. Vinel, and Y. Zhang. Attack detection and distributed forensics in machine-to-machine networks. *IEEE Network*, 30(6):49–55, November 2016.

[14] Y. Wang, T. Uehara, and R. Sasaki. Fog computing: Issues and challenges in security and forensics. In *2015 IEEE 39th Annual Computer Software and Applications Conference*, volume 3, pages 53–59, July 2015.

Sherenaz Al Haj Baddar is an Associate professor of Computer Science at the University of Jordan (JO), she earned her PhD degree in Computer Science from Kent State University in Ohio (USA) in 2009, then became an Assistant Professor of Computer Science at KASIT- The University of Jordan in 2009. In 2014, Dr. Al-Haj Baddar joined the Centro Interuniversitario di ricerca sull'Ingegneria delle Piattaforme Informatiche laboratory at the University of Genoa, Italy, to study anomaly detection in wireless networks. She co-authored the book entitled "Designing Sorting Networks: A New Paradigm" with Kenneth E. Batcher on novel strategies for designing faster sorting networks. Dr. Al-Haj Baddar research interests span anomaly detection in networks, as well as distributed computing. She has tutored more than 500 students at the University of Jordan.

Alessio Merlo received the MSc degree in computer science from the University of Genova, in 2005 and the PhD degree in computer science from the University of Genova, Italy, in 2010, where he worked on performance and access control issues related to Grid Computing. He is currently serving as an assistant professor at DIBRIS - University of Genoa. His currently research interests include performance and security issues related to web, distributed systems (Grid, Cloud), and mobile. He is involved as a member in program committees of international conferences (IFIP-SEC, AINA, ARES, HPCS, ...) and in the editorial board of an international journal (Journal of High Speed Networks).

Mauro Migliardi Mauro Migliardi is currently Associate Professor at the University of Padua, Adjunct Professor at the University of Genoa and a member of the Scientific Committee of the Center for Computing Platforms Engineering. He has been Research Associate at the Emory University and he has won the 2013 Canada-Italy Innovation Award. His main research interest is the engineering of secure, energy aware distributed systems. He tutored more than 100 among Bachelor, Master and PhD students at the Universities of Genoa, Padua and Emory, and he authored or co-authored more than 130 scientific papers.